Europäisches Patentamt | European Patent Office | Office européen des brevets

Bescheinigung | Certificate | Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. | Patent application No. | Demande de brevet n°

99126181.9

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

I.L.C. HATTEN-HECKMAN

DEN HAAG, DEN
THE HAGUE,
LA HAYE, LE

14/07/00

Europäisches Patentamt
European Patent Office
Office européen des brevets

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.:
Demande n°: 99126181.9

Anmeldetag:
Date of filing:
Date de dépôt: 30/12/99

Anmelder:
Applicant(s):
Demandeur(s):
International Business Machines Corporation
Armonk, NY 10504
UNITED STATES OF AMERICA

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
File futures

In Anspruch genommene Priorität(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat: State: Pays:	Tag: Date: Date:	Aktenzeichen: File no. Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:
/

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

D E S C R I P T I O N

File Futures

1. BACKGROUND OF THE INVENTION

1.1 FIELD OF THE INVENTION

The present invention relates to method and system for transferring data through a computer network. In particular, it relates to improvements in transferring large data amounts e.g., large files.

1.2 DESCRIPTION AND DISADVANTAGES OF PRIOR ART

The present inventions is applicable to any situation in which large amounts of data have to be transferred to a computer network. It has particular effects and improvements in the field of streaming 'new media' data such as audio and video data as it is increasingly required in business processes and with increasing acceptance of the Internet.

New media data extends traditional computer data formats into more natural data formats for the interaction of humans and computers by incorporating images, motion pictures, voice, audio, and video. Leading market, business, social, and technical indicators point to the growing importance of this digitally recorded content. Latest in 2003, new media data will eclipse structured data in sheer volume.

One of the key problems with new media data is transferring the usually huge amounts of content through a network. Normally, data is transferred using the store&forward paradigm, e.g. the complete content is transferred before anything is done with the data. A prominent implementation of this paradigm is the File Transfer Protocol (FTP), the standard way to transfer files throughout the World Wide Web.

For conventional data this works fine, as the amounts of data to be transferred are comparably small. For audio and video clips though, the latency time that passes between a request for rendering and the start of the rendering becomes unpractically long.

For this reason, in recent years a second paradigm called streaming has emerged. Streaming allows the rendering of the media to take place in parallel with the transfer of its content, which reduces latency times to a minimum. Streaming software always operates in pairs, a stream server pumping the data continuously through the network and a stream player receiving the data and rendering it.

Unfortunately, the way streaming is performed today has some serious side effects. First of all, the direct coupling of data transfer and doing something with the transferred data makes current streaming technology only available for special purposes, like rendering. Also, there are differences in handling streaming compared to store&forward, like the requirement of providing metafiles instead of the media itself, or different and proprietary security means.

Further, the correlation between stream server and renderer has led to proprietary protocol add-ons being transferred between them two for tactical reasons, practically preventing the player products of manufacturer A to work together with the stream server products of manufacturer B and vice versa.

Further, prior art streaming technology is restricted to stream data which are stored on the same computer device in which the stream server in use is residing. This, however, prevents streaming from being accomplished from any proprietary data server, as e.g., a DB2 database acting as source of the data stream. This reveals a considerable disadvantage as there is thus

a practical constraint to store business relevant data on the streamer hardware platform instead on a higher quality computer system with an increased degree of data security.

1.3 OBJECTS OF THE INVENTION

It is thus an object of the present invention to alleviate the disadvantages caused by proprietary streamer/player pairs.

2. SUMMARY AND ADVANTAGES OF THE INVENTION

These objects of the invention are achieved by the features stated in enclosed independent claims to which reference should now be made. Further advantageous arrangements and embodiments of the invention are set forth in the respective subclaims.

The inventional concept is able to transparently minimize latency times that come with traditional file transfer in a store&forward manner:

The present invention comprises basically to decouple the transfer and the rendering of media in a way that combines the strength, i.e., the advantages of the streaming paradigm with the flexibility and usability advantages of the store&forward paradigm.

The basic inventional principles are referred to herein as 'File Futures' and 'Future FTP' since an application, e.g., a renderer or player of new media data which uses the inventional principles can first initiate the transfer of the needed file through an inventional method replacing standard FTP and -while the file is being transferred - is enabled to already start working on the "future content" of the file currently being transferred.

This is basically achieved by separating the application logic from the transport logic. The transfer/transport logic is covered by an inventional transfer protocol whereas the player

application logic can issue standard file access statements like fread, fseek in order to access and render the streamed data.

This means that the invention consists of two parts, a Future client/server pair providing file transfer services as e.g., like FTP and an inventional Future File System Extension. These two parts communicate via an open interface which is able to be standardized. This implies indeed that the implementation of the Future FTP client/server pair and the Future File System Extension can be supplied by different parties.

The Future FTP client/server pair can use streaming technology and/or standard store and forward technology to perform the transfer. If streaming is used though, the necessary additional parameters such as required bandwidth may not show up in the Future FTP client interface. Instead, they can either be provided by a separate system management interface, for example a configuration file, or, they can be determined on the fly by interpreting the source file content. Optional qualities of service such as a bandwidth reservation through RSVP can be applied transparently.

The Future FTP client interface follows the standard, and the extension necessary for the interaction with the Future File System Extension is able to be standardized, too.

In contrary to normal, i.e., prior art files, the inventional File Futures allow the content of a file to be read while parts of it are still transferred, therefore hiding the fact that the file is not completely available yet. For this reason, a file future knows it's related Future FTP Client instance through a callback interface mechanism.

For the scope of the present invention it is assumed that while the transfer of a future file is in progress no application is

allowed to write to that file. Once the transmission is completed the relationship of the Future File System Extension to the Future FTP client for that session is dissolved and later access requests to the transferred file are treated as usual prior art file requests. This implies that write access to that file is allowed from that time on.

The File Future File System Extension provides a mechanism to allow recovery from broken FTP transfers as otherwise an application cannot be sure if a transfer really completed due to it's asynchronous nature. In other words, the Future File System Extension is able to reinitiate transfers in effect to guarantee that a transfer completes successfully.

File Futures are intended as a replacement for all means of FTP file transfer paradigms and have the advantage of transparently minimizing latency times for accessing the required file content. It can therefore be used in all application areas where applications do not work on the complete file content, but instead only use sections of a file at a time. Examples for these kinds of applications are:

Rendering applications like video and audio players. Using the File Future technology will make proprietary stream server products obsolete to a high degree, as all media players can render using standard file access.

CAD/CAM applications profit because they're usually designed in a way to work on smaller portions of a file at a time due to huge size of CAD/CAM files. Therefore, using the inventional File Future technology will give CAD/CAM products advantages similar to the advantages which media renderers gain today from using streaming.

The inventional technology can be advantageously exploited to integrate proprietary stream servers into an enterprise network.

3. BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and is not limited by the shape of the figures of the accompanying drawings in which:

Fig. 1 is a schematic block diagram showing the most essential elements in a distributed system which is used for the present invention according to one preferred aspect thereof as well as the basic activities during the inventional file transfer method,

Fig. 2A,B is a schematic block diagram showing the most essential steps according to one aspect of the inventional file transfer method, and

Fig. 3 is a schematic block diagram of a variation of fig. 1, showing a system structure which is also able to be realized according to a second preferred aspect of the inventional file transfer method.

4. DESCRIPTION OF THE PREFERRED EMBODIMENT

With general reference to the figures and with special reference now to fig. 1 an insurance company has a data server 10 using a conventional file system 12 in order to store business-relevant data. Some of the data is stored in form of video films consuming large amounts of bytes. As, e.g., several hundreds of megabytes. The data server 10 is connected to a stream server 16 via an inventional transfer component 14 and a high speed data transmission line 15.

Said inventional software component 14 acts as a sender component for transferring files of the file system 12 to some predetermined target computer systems one of which is denoted as 16. As said component 14 forms part of the inventional file future concepts it is referred to herein as future FTP server and

abbreviated as FFTP server 14. Said stream server 16 comprises a stream server application 20 which is implemented together with the stream server's file system 24 in one and the same hardware platform. In this case a UNIX workstation is used.

Further and according to some basics of the inventional concepts there is implemented an additional program component which cooperates with the file system 24 and acts as a file system extension 22. As said additional software component is concerned with the inventional provision of the above mentioned file futures said component 22 is referred to herein as future file system extension and abbreviated as FFSE.

Said FFSE component 22 is arranged to cooperate with a further software component 18 which serves as a file transfer client program. As it forms part of the inventional file future concepts it is referred to herein as Future FTP client and abbreviated herein as FFTP client. Said FFTP client 18 acts as a receiver program for files which are transferred from the future FTP server associated with the data server 10 mentioned above.

Further, a personal computer 30 which is associated to some staff of said insurance company, acts as an end-user client in relation to the stream server 16 mentioned above. Said end-user PC is provided with some player program which is capable to render video data and/or audio data which are received from the stream server 16. As reveals from the drawing the relevant hardware components form a 'triangle' comprising the data server 10, the stream server 16 and the end-user PC 30 as angles thereof.

With additional reference now to **fig. 2 A and B** the most essential steps and the control flow according to one aspect of the inventional file transfer method are described in more detail next below and using the configuration scheme depicted in fig. 1.

Said application 20 first initiates a Future FTP transfer, passing the file name of the remote source file and the file name

of the local target file to the future ftp client. This is done in the same syntax as in standard FTP, see step 205.

Then, in a step 210 said FFTP client 18 creates a local target file using the standard file system 24. Further, step 215, it registers a prior art callback with the help of the inventional Future File System Extension for the control flow to be expected, e.g., via passing a process ID or a port number. After that, it initiates the asynchronous transfer of the file content with the Future FTP server, which results in the bytes being transferred over the network, step 220. Finally, it returns to the application.

The streamer application 20 opens the target file, step 225, and is able to access the content while it is still being transferred. In general, some calls received by the file system 24 are routed to the inventional Future File System Extension 22 for special behavior.

This can be realized for example by the technique of stacking file systems or using the XDSM technology. Further, it is possible to enrich the code of a file system which is open to application programmers as it is e.g., the LINUX file system with a kind of wrapper logic which implements the above described mechanism, i.e., to catch the relevant file access commands issued by the application and to route it to the FFTP client for further processing, or simply pass it to another FFTP process running in the same computer device or, in general, to any location as it is required for the actual streaming process.

As long as the file transfer between said FFTP server component 14 and the FFTP client 18 is not yet completed, decision 230, as it can be the case with a large amount of data to be transferred two cases must be seperated when the end-user associated with the PC ststion 30 issues some file access commands to the streaming

server application 20 referring to the file being rendered by the player 32. In any case those commands are routed, step 235, to the inventional FFSE component 22 in order to be managed by it.

If the application issues a read - left branch of step 235 and a decision 240 yields that all bytes for this request are already available, it immediately returns the requested bytes to the application, step 245. If the content is not available, it waits - step 250 - for the content to arrive and then returns it, step 245.

If the application issues a seek as it is depicted in the right branch after step 235, the Future File System Extension will order the Future FTP Client to reposition the transfer stream and return immediately to the application, step 255. This causes the asynchronously running file transfer to start transferring from the new requested position at the next possible opportunity. Thus, streaming is continued with the data portions the end-user wished to be rendered by his seek command, step 265.

Of course, this means that the file pointer on the target file system is also repositioned to the new location, step 260, which results in "holes" in the target file.

If the filling of the holes is desired as it is expected to be in most cases this can be handled by the FFTP client 18 which recursively fills them automatically once it has transferred to the file end, see the branch back to step 220.

With reference now to **fig. 3** some preferred variation of the basic configuration according to a second, preferred aspect of the present invention depicted in fig. 1 is described in more detail.

As reveals from the drawing all essential functionality which was depicted to be implemented in the stream server 16 in fig. 1 can also be implemented in the end-user PC 30 itself. Thus, all file access commands as they are in particular read and seek commands are issued by the player 30 directly to to to the FFSE component 22 for being handled in cooperation with the FFTP client 18 and FFTP server components 14.

According to this aspect the stream server 16 becomes obsolete which saves costs an has the effect that the data to be rendered at the end-user PC 30 may be transferred directly from the data server 10 through the network to the end-user PC 30. Thus, the data need not to be present physically at a third location in the network which is advantageous when business-critical data are to be rendered. Further, any player program can be used with this approach as the inventional Future File concept can be implemented offering an open standard covering all players being commercially available. Thus, the disadvantageous effects of prior art required proprietary pairs of stream server/player are eliminated.

In general, the inventional File Futures principles can be positioned to both streaming technology and store&forward technology. Compared to streaming technology, File Futures have the advantage to be absolutely transparent to the application. In other words, the application still works on a strict file paradigm basis and does need to be reprogrammed to be enabled to know that the complete file is not available at the time it tries to access it.

Compared to traditional store&forward technology, the inventional File Futures have the advantage to minimize the latency time, which is a use inhibitor for store&forward mechanisms when large files are transferred. In effect this means that the inventional concepts of File Futures combine the strength, i.e., the advantages of both paradigms, streaming and store&forward.

The inventional approach described above can be advantageously combined with the natural 'inner' structure of many data sources having huge contents, like books comprising many images, photo catalogues, music pieces, films or large business data files in for example banking or CAD applications which often use a table of contents for the user to navigate quickly in the data. Thus the end-user will often have the possibility to first see the table of contents for a selection of the desired data. Then a file acces acommand like the above described 'seek' will usually follow - and will be supported according to the present invention with minimized latency. Basically the same can be done with a conventional table of indexes, as well.

Thus, the present invention represents a large step forward for improving the performance of sessions in the World Wide Web where in many cases the user's strongest desire is to retrieve quickly what he searches for.

In the foregoing specification the invention has been described with reference to a specific exemplary embodiment thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are accordingly to be regarded as illustrative rather than in a restrictive sense.

The present invention can be realized in hardware, software, or a combination of hardware and software. A Future File Transfer tool according to the present invention can be realized in a centralized fashion in one computer system, or in a distributed fashion where different elements are spread across several interconnected computer systems. Any kind of computer system or other apparatus adapted for carrying out the methods described herein is suited. A typical combination of hardware and software could be a general purpose computer system with a computer

program that, when being loaded and executed, controls the computer system such that it carries out the client or server specific steps of the methods described herein.

The present invention can also be embedded in a computer program product, which comprises all the features enabling the implementation the respective steps of the methods described herein, and which - when loaded in one or more computer systems - is able to carry out these methods.

Computer program means or computer program in the present context mean any expression, in any language, code or notation, of a set of instructions intended to cause a system having an information processing capability to perform a particular function either directly or after either or both of

the following

a) conversion to another language, code or notation;

b) reproduction in a different material form.

C L A I M S

1. A client-server based file transfer method comprising the steps of:

   keeping at the client computer system (16, 30) at least a portion of a file received from a server (10) ready for being accessed by an application program (20, 32) while said file is being transferred between said server (10) and said client (16, 30),

   fulfilling (225, 235, 245) application program-initiated requests for accessing specified portions of said file while said file is being transferred.

2. The method according to claim 1 in which said step of fulfilling application program-initiated requests comprises the steps of:

   providing (245) requested contents of said file portion to the application program (20, 32) if said contents have already been received from the data server (10),

   otherwise re-initiating (255, 260, 265) the file transfer to be continued beginning with a file position requested from the application program ((20, 32).

3. Use of the method according to one of the preceding claims for rendering data on a client computer system (16, 30).

4. The use of the method according to the preceding claim for directly transferring data between a server (10) and an end-user client (30).

5. The use of the method according to claim 3 or 4 in which new media data is transferred by streaming the data from the

server (10).

6. A client-server based file transfer method comprising the steps of:

streaming client-requested file information to a client computer system (16, 30) in portions according to specifications issued by said client (16, 30).

7. The method according to the preceding claim in which said step of streaming is performed by

streaming the requested file sequentially skipping portions of the file already streamed before.

8. A client computer system having means for performing the steps of a method according to one of the preceding claims 1 to 5 and acting as a client system with reference to said data server system (10).

9. The client computer system (16) according to the preceding claim having means for acting as a server system with reference to a remote end-user associated computer system.

10. Data server computer system having means for performing the steps of a method according to one of the preceding claims 6 to 7.

11. A computer program for execution in a data processing system comprising computer program code portions for performing respective steps of the method according to anyone of the claims 1 to 7.

12. A computer program product stored on a computer usable medium comprising computer readable program means for causing a computer to perform the method of anyone of the claims 1 to 7.

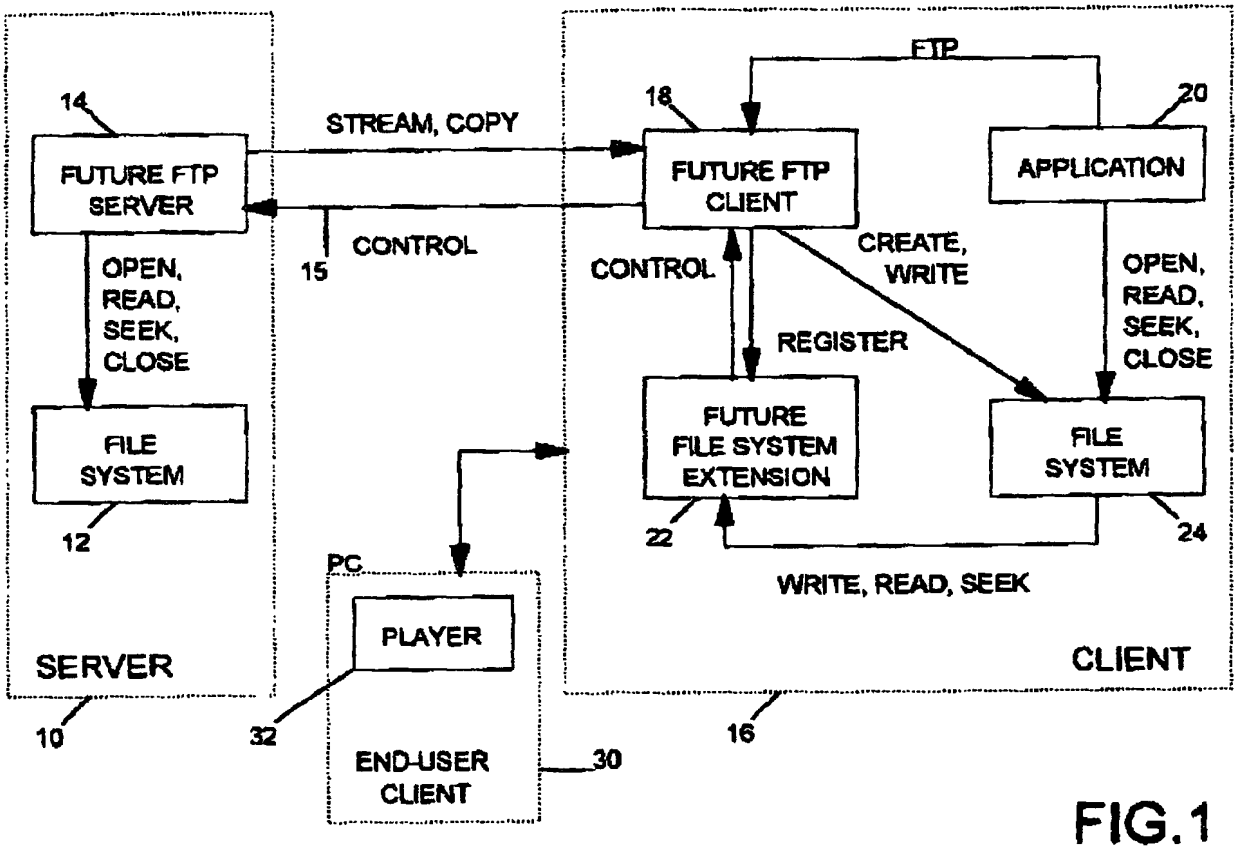
FTP
14
STREAM, COPY
18
20
FUTURE FTP SERVER
FUTURE FTP CLIENT
APPLICATION
CONTROL
15
CONTROL
CREATE, WRITE
OPEN, READ, SEEK, CLOSE
OPEN, READ, SEEK, CLOSE
REGISTER
FILE SYSTEM
12
FUTURE FILE SYSTEM EXTENSION
FILE SYSTEM
22
24
WRITE, READ, SEEK
PC
PLAYER
SERVER
CLIENT
10
32
16
END-USER CLIENT
30

FIG.1

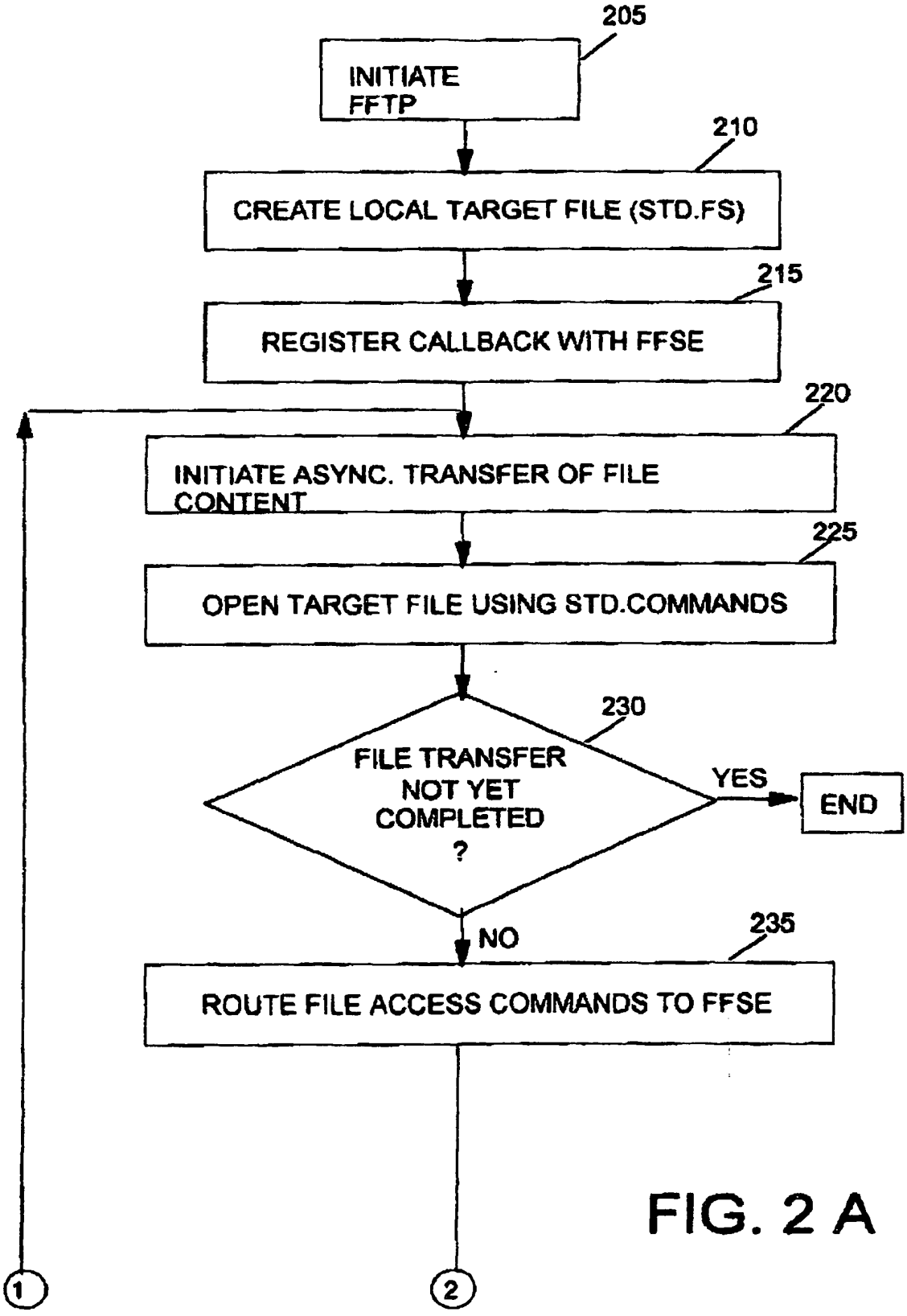
205
INITIATE FFTP
210
CREATE LOCAL TARGET FILE (STD.FS)
215
REGISTER CALLBACK WITH FFSE
220
INITIATE ASYNC. TRANSFER OF FILE CONTENT
225
OPEN TARGET FILE USING STD.COMMANDS
230
FILE TRANSFER NOT YET COMPLETED ?
YES
END
NO
235
ROUTE FILE ACCESS COMMANDS TO FFSE
1
2

FIG. 2 A

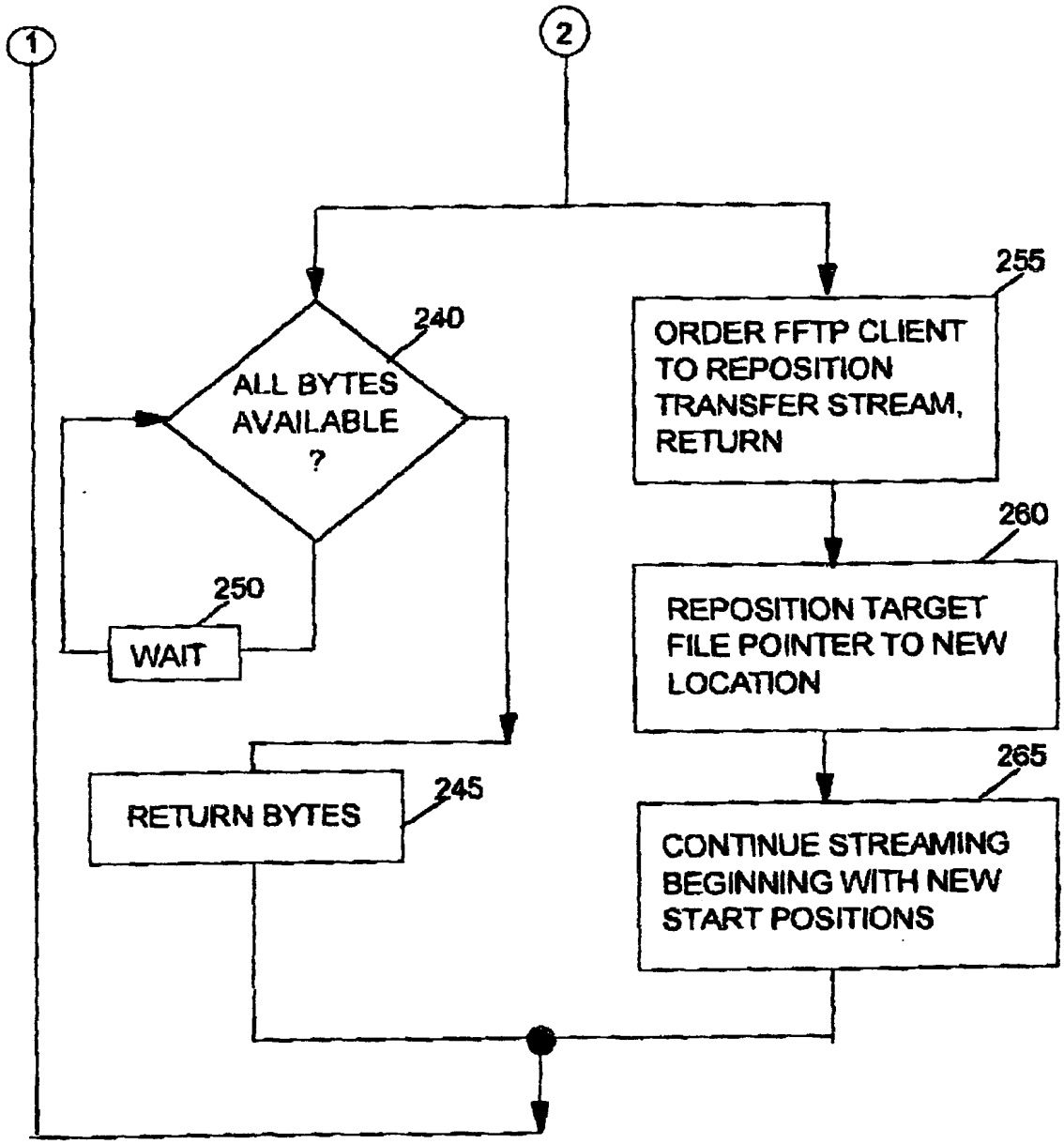
1
2
240
ALL BYTES AVAILABLE ?
250
WAIT
RETURN BYTES
245
255
ORDER FFTP CLIENT TO REPOSITION TRANSFER STREAM, RETURN
260
REPOSITION TARGET FILE POINTER TO NEW LOCATION
265
CONTINUE STREAMING BEGINNING WITH NEW START POSITIONS

FIG. 2B

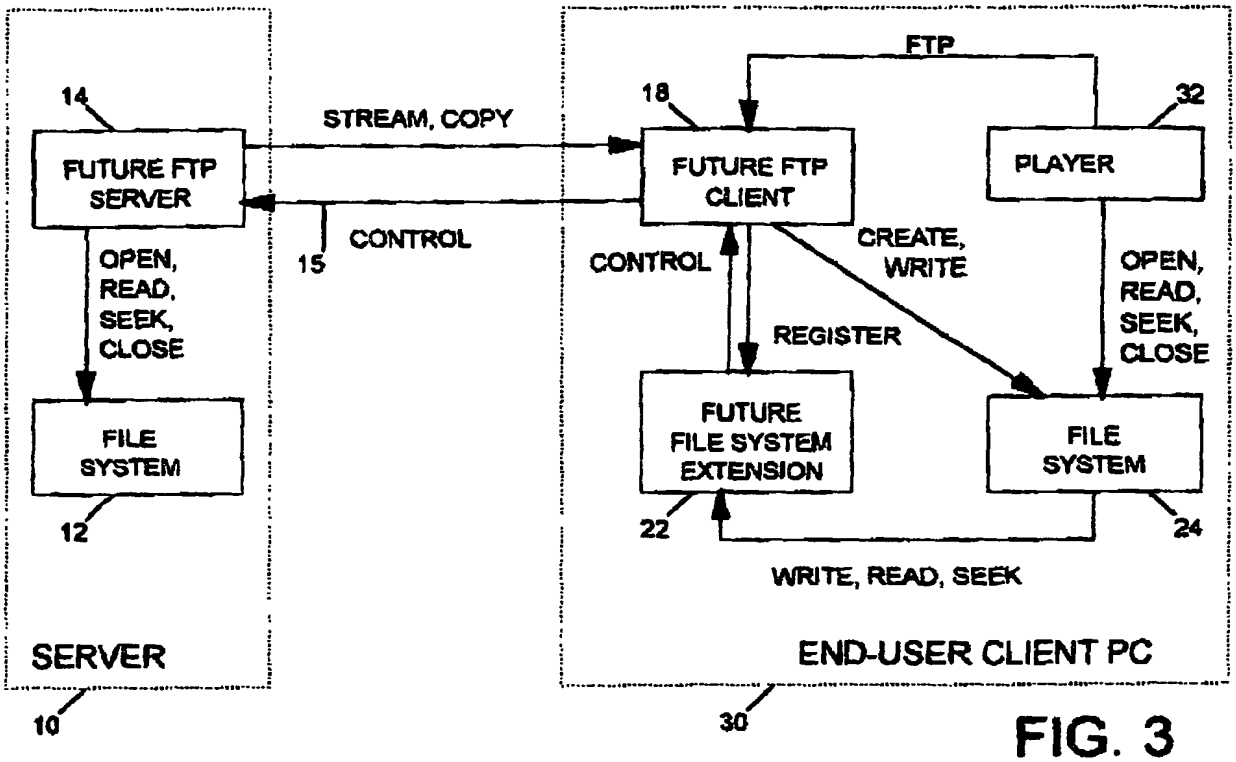
14
FUTURE FTP SERVER
STREAM, COPY
CONTROL
15
OPEN, READ, SEEK, CLOSE
FILE SYSTEM
12
SERVER
10
FTP
18
FUTURE FTP CLIENT
32
PLAYER
CONTROL
CREATE, WRITE
OPEN, READ, SEEK, CLOSE
REGISTER
FUTURE FILE SYSTEM EXTENSION
FILE SYSTEM
22
24
WRITE, READ, SEEK
END-USER CLIENT PC
30

FIG. 3

Leerseite

A B S T R A C T

The present invention relates to method comprising improvements in transferring large data amounts e.g., large files associated with 'new media' like audio and video data which are streamed through the network.

The present invention comprises basically to decouple the transfer and the rendering of data in a way that combines the strength, i.e., the advantages of the streaming paradigm with the flexibility and usability advantages of the store&forward paradigm.

This is basically achieved by separating the application logic from the transport logic. The transfer/transport logic is covered by an inventional transfer protocol whereas the player application logic can issue standard file access statements like fread, fseek in order to access and render the streamed data. (Fig. 1)

A B S T R A C T

The present invention relates to method comprising improvements in transferring large data amounts e.g., large files associated with 'new media' like audio and video data which are streamed through the network.

The present invention comprises basically to decouple the transfer and the rendering of data in a way that combines the strength, i.e., the advantages of the streaming paradigm with the flexibility and usability advantages of the store&forward paradigm.

This is basically achieved by separating the application logic from the transport logic. The transfer/transport logic is covered by an inventional transfer protocol whereas the player application logic can issue standard file access statements like fread, fseek in order to access and render the streamed data. (Fig. 1)